

# Support Tools for PVS Theorem Proving: A Quest for Knowledge

Ben L. Di Vito

NASA Langley Research Center  
Formal Methods Team

`b.l.divito@larc.nasa.gov`

phone: (757) 864-4883

fax: (757) 864-4234

`http://shemesh.larc.nasa.gov/~bld`

23 October 2003

## Why Johnny Can't Prove

Tools such as PVS have some great capabilities, but they...

- Are woefully under-educated
  - Have the mathematical knowledge of a middle school student
- Force us to spend too much time teaching them
- Make us repeat the exercise too often
  - Only limited features aimed at reuse
  - Create a large drag on productivity

These circumstances will limit the uptake of formal methods.

What we need to do:

- Send PVS to high school
  - Plus one or two years of college
- Make sure the lessons are memorized

## Enter Johnny Theoremseed

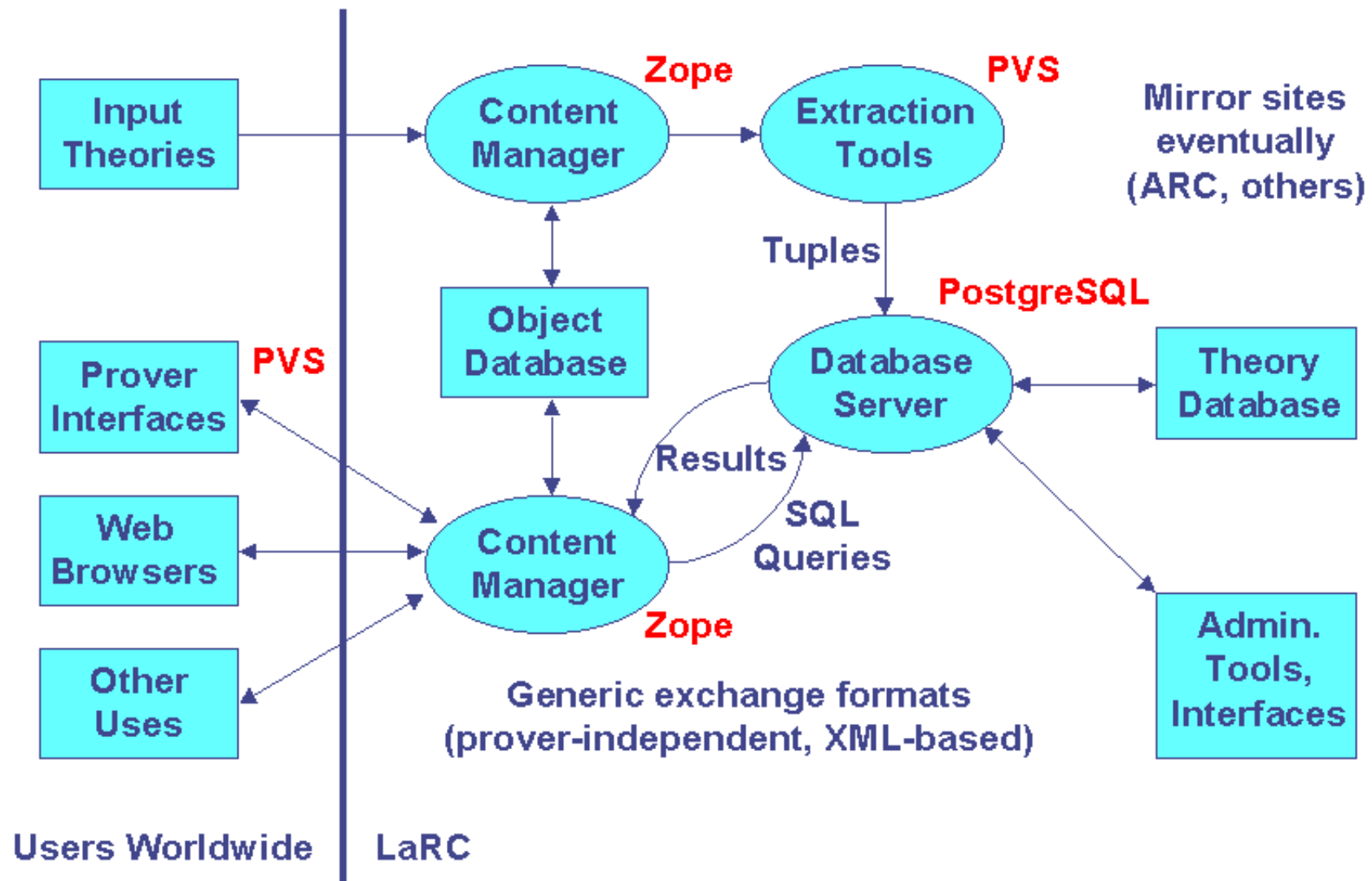
We wish spread deductive/mathematical knowledge by hosting a dedicated web server and providing a specialized set of services to PVS users:

- A database of deductive and mathematical artifacts
- A web-based interface mechanism to:
  - Issue queries against the database
  - Submit new content for inclusion in the database
- A client module to complement PVS
  - Offers proof-side assistance during prover sessions
  - Automates the discovery and acquisition of relevant theorems
- An extensible platform for implementing future services
  - A programmatic interface (API) for invoking services

We design the service and tools in hopes of attracting contributions from the PVS community

- Users benefit from what we offer
- They are motivated to reciprocate
- A passive collaboration process results

# Client-Server Architecture



# Hypatheon

| [Home](#) | [Introduction](#) | [Obtaining and Using the PVS client](#) | [Submit content](#) | [Query the database](#) |

---

## Database of Deductive Knowledge

Welcome to the Hypatheon database of deductive knowledge. Here you will find a collection of mathematics formalized using SRI's [PVS](#) language and tools. This database service is provided and maintained by the [formal methods team](#) at [NASA Langley Research Center](#). It has been developed under NASA's Engineering for Complex Systems Program.

The following information and services are available:


- [Introduction](#)
- [Obtaining and Using the PVS client](#)
- [Submit content](#)
- [Query the database](#)

The Hypatheon [development team](#) welcomes your [feedback and suggestions](#).

---

Curator and Responsible NASA Official: [Ben Di Vito](#)  
[larc privacy statement](#)

last modified: October 22, 2003 3:39 pm GMT-4

Note: The  to external site tag identifies links that are outside of the NASA domain.



# Hypatheon

| [Home](#) | [Introduction](#) | [Obtaining and Using the PVS client](#) | [Submit content](#) | [Query the database](#) |

---

## Query the database

The database may be searched directly from the following input forms. Tabular results are returned and displayed by your browser. A PVS client is also available for proof-side searching.

Search for Declarations:

- [Search for lemmas that refer to functions](#)
- [Search for lemma names by pattern](#)
- [Search for function names by pattern](#)
- [Search for functions that refer to other functions](#)

Search for Theories:

- [Search for theory names by pattern](#)
- [Find theories required by given theory](#)
- [Find transitive closure of theories that are required by a given theory](#)
- [Find theories that depend on given theory](#)
- [Find transitive closure of theories that depend on given theory](#)

Search for Libraries:

- [List information about libraries](#)

Display database information:

- [Show database summary statistics](#)
- 

October 22, 2003 3:48 pm GMT-4

# Hypatheon

[| Home](#) | [Introduction](#) | [Obtaining and Using the PVS client](#) | [Submit content](#) | [Query the database](#) |

---

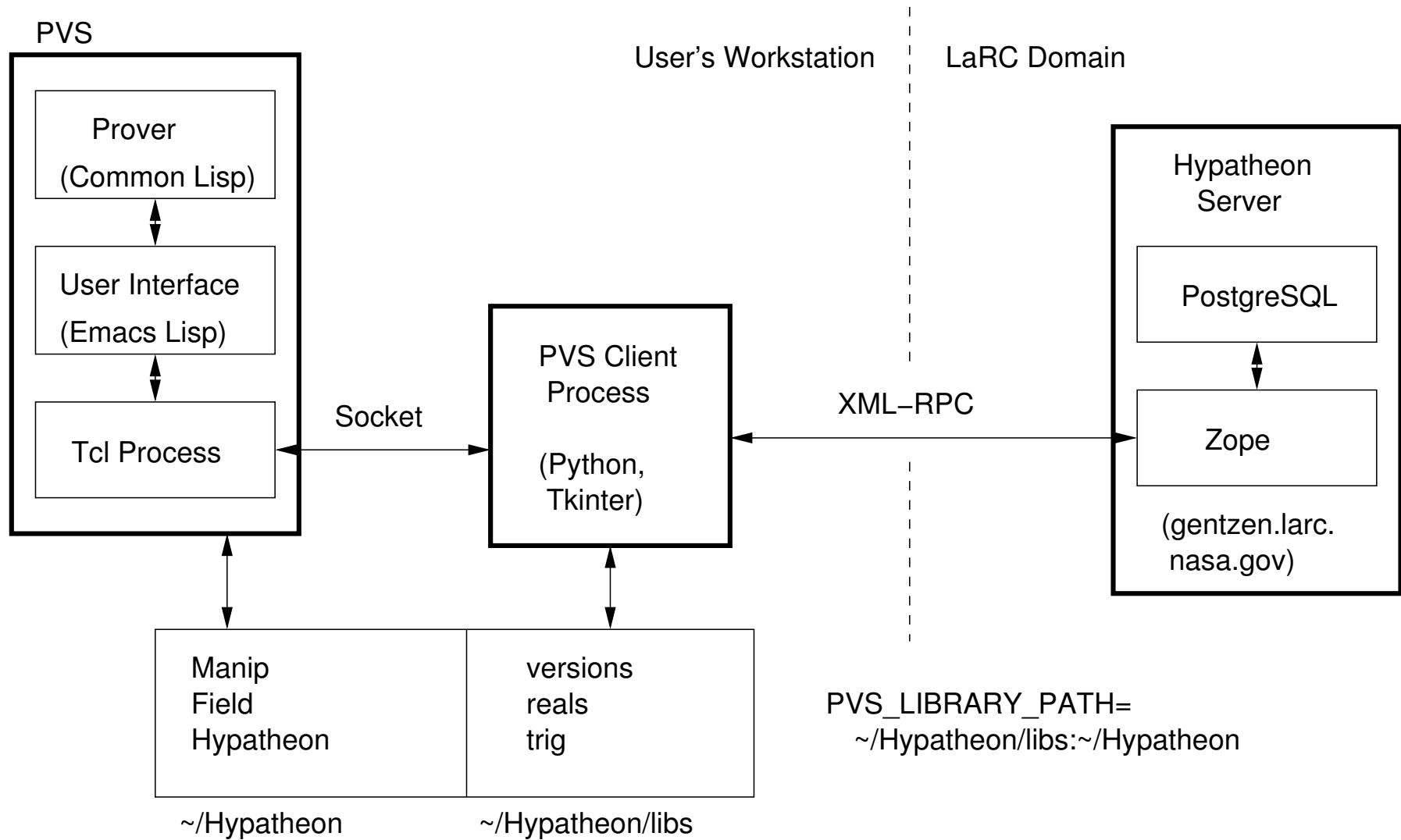
13 records found.

Declaration	Theory	Library
<a href="#">Law_Cosines</a>	law_cosines	trig
<a href="#">sq_dist_is_dist_sq</a>	position	vectors
<a href="#">dist_triangle</a>	position	vectors
<a href="#">law_cosines</a>	law_cos_pos2D	vectors
<a href="#">law_cosines_alt</a>	law_cos_pos2D	vectors
<a href="#">law_cosines_bnd</a>	law_cos_pos2D	vectors
<a href="#">sq_dist_is_dist_sq</a>	position2D	vectors
<a href="#">dist_triangle</a>	position2D	vectors
<a href="#">sq_dist_is_dist_sq</a>	position3D	vectors
<a href="#">dist_triangle</a>	position3D	vectors
<a href="#">law_cosines</a>	law_cos_pos3D	vectors
<a href="#">law_cosines_alt</a>	law_cos_pos3D	vectors
<a href="#">law_cosines_bnd</a>	law_cos_pos3D	vectors

---

Results produced by Hypatheon on October 22, 2003 3:53 pm GMT-4.

# Design of PVS Client





# PVS Client Module

The screenshot displays the PVS Client Module interface, which is divided into three main windows:

- PVS@air57 <2>**: The main editor window showing a proof script. The script includes a goal `(1) FORALL (a: real): cos(a) = sin(a + PI / 2)` and several rewrite rules. The current state shows the goal `(1) cos(a/1) = sin(a/1 + PI / 2)` and the application of the `cos_PI2` lemma, resulting in `(1) cos(a/1) = cos(a/1) * 1 + sin(a/1) * 0`.
- Hypatheon Client for PVS**: A window for searching for functions or formulas. It contains a "Query Control Panel" with a "Functions:" input field (containing `cos /`) and a "Formula:" input field. Both have "Submit" buttons. A "Clear Entries" button is also present.
- Query Window <3>**: A window showing the results of a query. The query is `cos /`. The results list includes `cos_half_zeroes2`, `cos_le_0`, `cos_lt_0`, `cos_minus_3PI2`, `cos_minus_PI2`, `cos_PI2`, `cos_PI2_minus`, `cos_PI3`, `cos_PI4`, and `cos_PI6`. The `trig_basic` lemma is highlighted. The window also shows "Query: 61 results found" and buttons for "Lemma", "Use", and "Rewrite".

# Using Hypatheon

Visit [http://gentzen.larc.nasa.gov:8080/hypatheon\\_dev](http://gentzen.larc.nasa.gov:8080/hypatheon_dev)

- Available only within LaRC domain
- Queries helpful during specification writing as well as proving
- Submissions are standard PVS theories (whatever typechecks against the database)
- Download and installation required for client package
- Instructions are found on the download page
- Client maintains directory of libraries on user's workstation
- Setup includes Manip, Field and LaRC libraries
- Provides an isolated PVS environment for testing Hypatheon

```
sqrt: THEORY
BEGIN

  IMPORTING sq, sqrt_exists
  . . .

  sqrt_gt1      : LEMMA  nnx > 1 IFF sqrt(nnx) > 1

  sqrt_ge1      : LEMMA  nnx >= 1 IFF sqrt(nnx) >= 1

  sqrt_plus_le :
    LEMMA  sqrt(nnx+nnny) <= sqrt(nnx) + sqrt(nny)

  sqrt_cauchy  :
    LEMMA FORALL (a,b,c,d: real):
      a*c + b*d <=
        sqrt(sq(a)+sq(b)) * sqrt(sq(c)+sq(d))

  %% ===== Theory extension =====

  sqrt_crazy   : LEMMA irrational(sqrt(2))

  %% ===== Theory extension =====

  still_crazy  : LEMMA irrational(sqrt(3))

END sqrt
```

## Programmatic Interface

Design and implement third database interface.

- Provide API for generic database services
- Support arbitrary SQL queries
- Encourage new users and uses
  - Both researchers and advanced practitioners
  - Possibly of interest to non-PVS communities
- Enable custom proof automation for specialized domains
- Potentially useful to mathematical knowledge management (MKM) groups
  - Import/export data to other notations/formalisms

## Advanced Queries

Next step in query development aims for greater automation.

- Need heuristics for ranking search results
- Try automatically selecting suitable lemmas
- Imagine a choice function  $S : \text{proof\_state} \times \text{database} \rightarrow \langle \text{lemmas} \rangle$
- Existing library proofs available as a training dataset for  $S$ 
  - Similar to a large curve-fitting problem
  - Over 3000 data points to draw on
- Goal is to implement heuristics that consistently pick the “correct” lemma or at least rank it highly
- Investigate feasibility of finding variable instantiations
- Efficiency of  $S$  is an issue
  - Augment database to ensure adequate performance
  - Build auxiliary database tables with added relationships
  - Precompute (portions of)  $S$  as necessary

## Observation: Use of Lemmas in Library Proofs

Library	Commands:	USE	LEMMA	REWRITE	
prelude		53	388	115	
bitvectors		0	0	4	
finite_sets		12	25	83	
algebra		0	23	8	
analysis		100	121	149	
arrays		0	34	44	
bags		12	7	13	
calculus		2	78	64	
digraphs		4	91	44	
div		0	60	8	
fixedpoints		0	3	11	
graphs		5	230	107	
mod		0	29	23	
nat_funs		3	45	16	
number_theory		0	117	62	
powersets		5	48	29	
reals		5	68	105	
series		0	98	36	
trig		3	328	417	
vectors		0	0	3	
Totals		204	1793	1341	⇒ 3338

## Exploiting Proof Information

Having fully mechanical proofs creates new targets of opportunity.

- Add proof artifacts to database
  - Complete PVS proofs
  - Break into steps to uncover structure
  - Relate proof steps to other lemmas/proofs
- Identify patterns and idioms
- Support searches based on proof content
- Enable proof cloning
  - Help users to clone their own proofs
  - Help users to find and clone others' proofs
  - Semi-automatic tailoring (e.g., substitutions:  $f \rightarrow g$ )
  - Example: 2D vs. 3D vector theories

# Data Mining Opportunities

After a sufficient body of knowledge is collected, it can be mined for new information.

- Off-line analysis of database contents
- Derive measures of coverage, effectiveness
- Discover patterns in lemmas, theories, proofs
- Identify areas needing attention
  - Automatically search for gaps, weak coverage
  - Also search for needless redundancy
  - Suggest new lemmas based on global analysis results
- Consider collecting usage statistics
  - Would help tune database and improve search results
  - Users' privacy concerns could make this tricky

# Plans

- Continue to refine current prototype
- Prepare for external server rollout
  - Late 2003 / early 2004
- Conduct performance/capacity testing
- Goal is for server to support:
  - 1 K libraries
  - 10 K theories
  - 100 K function definitions
  - 1 M theorems (formulas)
- Make improvements based on user feedback
- Add proof handling features
- Develop advanced query capabilities
- Pursue data mining opportunities